

SACLANTCEN Technical Memorandum No. 173

AD735994



SACLANT ASW
RESEARCH CENTRE

SACLANTCEN

Technical Memorandum No. 173

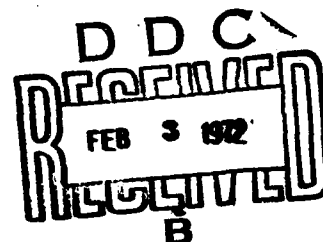


To DDC

A COMPUTER PROGRAM FOR CONTOUR MAPPING

by

JENS M. HOVEN



gr

15 OCTOBER 1971

ESRO

NORTH
ATLANTIC
TREATY
ORGANIZATION

VIALE SAN BARTOLOMEO 400
I-19026 - LA SPEZIA, ITALY

This document is unclassified. However, the information it contains is published subject to the conditions of the legend printed on the inside cover: short quotations from it may be made in other scientific publications if credit is given to the author(s) and to SACLANTCEN; requests for other reproduction, except in official NATO publications, should be addressed to the Director, SACLANTCEN.

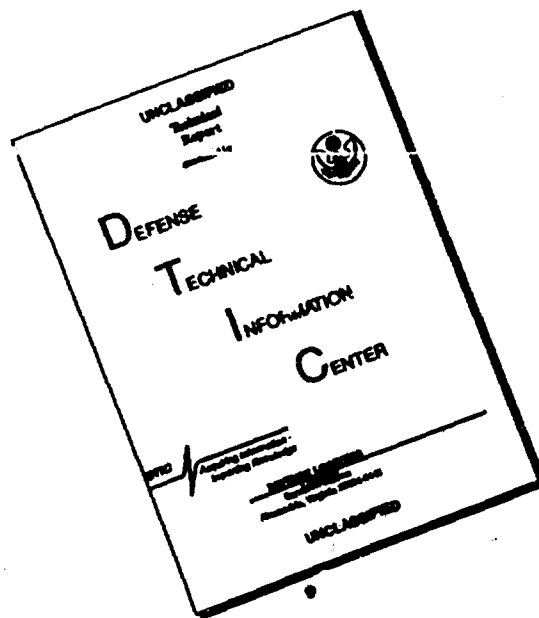
DISTRIBUTION STATEMENT A

Approved for public release
Distribution Unlimited

Reproduced by
NATIONAL TECHNICAL
INFORMATION SERVICE
Springfield, Va. 22151

20

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

SACLANTCEN
TECHNICAL MEMORANDUM NO. 173

NORTH ATLANTIC TREATY ORGANIZATION
SACLANT ASW RESEARCH CENTRE
Viale San Bartolomeo 400
I 19026 - La Spezia, Italy

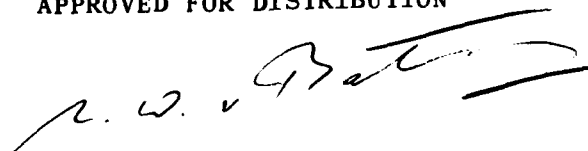
A COMPUTER PROGRAM FOR CONTOUR MAPPING

by

Jens M. Hovem

15 October 1971

APPROVED FOR DISTRIBUTION

A handwritten signature in dark ink, appearing to read 'M.W. van Batenburg', with a horizontal line drawn through the middle of the signature.

Ir M.W. van Batenburg
Director

ACCESSION IN		
CFSTI	WHITE SECTION <input checked="" type="checkbox"/>	
ROC	DIFF SECTION <input type="checkbox"/>	
MARKED	<input type="checkbox"/>	
JUSTIFICATION		
BY		
DISTRIBUTION/AVAILABILITY CODES		
DIST.	AVAIL. and/or SPECIAL	
A		

This document is released to a NATO Government at the direction of the SACLANCEN subject to the following conditions:

1. The recipient NATO Government agrees to use its best endeavours to ensure that the information herein disclosed, whether or not it bears a security classification, is not dealt with in any manner (a) contrary to the intent of the provisions of the Charter of the Centre, or (b) prejudicial to the rights of the owner thereof to obtain patent, copyright, or other like statutory protection therefor.

2. If the technical information was originally released to the Centre by a NATO Government subject to restrictions clearly marked on this document the recipient NATO Government agrees to use its best endeavours to abide by the terms of the restrictions so imposed by the releasing Government.

ACCESSION IN		
CFSTI	WHITE SECTION <input type="checkbox"/>	
ROC	DIFF SECTION <input type="checkbox"/>	
MARKED	<input type="checkbox"/>	
JUSTIFICATION		
BY		
DISTRIBUTION/AVAILABILITY CODES		
DIST.	AVAIL. and/or SPECIAL	

TABLE OF CONTENTS

	<u>Page</u>
ABSTRACT	1
INTRODUCTION	2
1. PRINCIPLE OF THE METHOD	3
2. PROGRAM DESCRIPTION	5
3. METHOD OF USE	8
4. EXAMPLE OF APPLICATION	10
CONCLUSIONS	12
REFERENCES	13
APPENDIX A	
LISTING OF AN ALGOL PROCEDURE FOR CONTOUR MAPPING	14

List of Figures

1. Representation of data matrix.	4
2. Possible ways of contouring at a saddle point.	7
3. Example of a contour map.	11

A COMPUTER PROGRAM FOR CONTOUR MAPPING

by

Jens M. Hovem

ABSTRACT

Contour mapping is frequently an attractive possibility for the graphic representation of two-dimensional functions or numerical values. This memorandum describes a simple computer program for the drawing of contours. The principles of the method applied are described and a program listing (an ALGOL procedure) is given in an appendix.

INTRODUCTION

In all technical and scientific work the choice of the method to be used for the graphic display of numerical results is most important. If correctly chosen it may greatly facilitate the interpretation of the results; on the contrary a badly chosen method may render the results almost useless.

When the results to be displayed are in the form of a function of two variables, $f(x, y)$, contour mapping of this function is often a very attractive way of representation. Contour mapping consists in drawing lines (contours) in the xy plane defined by the equation $f(x, y) = \text{constant}$. By selecting a set of different values for the constant, a representation of the function is obtained that may be particularly easy to understand.

The purpose of this paper is to describe a computer program for the drawing of contour lines that has been developed at SACLANTCEN. The program works on a matrix $A[k, p]$ representing the sample values of $f(x, y)$. It is assumed that the sampling is equidistant and that the sampling density is high enough to ensure that simple linear interpolation will determine the positions of the contour lines with sufficient precision.

The program was developed for use in connection with the processing of acoustic signals and an example of such a use is presented.

1. PRINCIPLE OF THE METHOD

Assume that the function $f(x, y)$ is defined in a rectangular domain of x and y and that it is sampled at regular intervals Δx and Δy . These samples can be assigned to a matrix $A[k, p]$, such that

$$A[k, p] = f(k \Delta x, p \Delta y)$$

defined for $k = 0, 1, \dots, N-1$ and $p = 0, 1, \dots, M-1$.

As a way of representation one can imagine a rectangular grid, in the xy plane, with the values of the matrix $A[k, p]$ associated to the nodes of the grid [Fig. 1]. By drawing lines between adjacent nodes the range of interest in the xy plane is thus divided into $(N-1)(M-1)$ cells with sides Δx and Δy .

Assuming that the sampling density is high enough to position the contours with a one cell precision, the drawing of a given contour line can be divided into two steps [Ref. 1]:

(1) Testing all pairs of adjacent matrix elements to find which cell and cell sides are intersected by the given contour line.

(2) Connecting the intersections by drawing lines.

Step 2 will necessarily require some interpolation in order to determine precisely the route of the contour. With the conditions of high sampling density, step 1 will however not require any interpolation and can therefore be made independently of step 2.

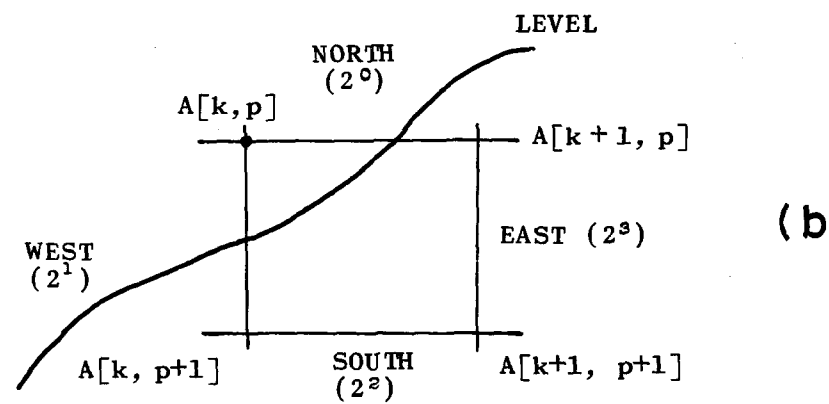
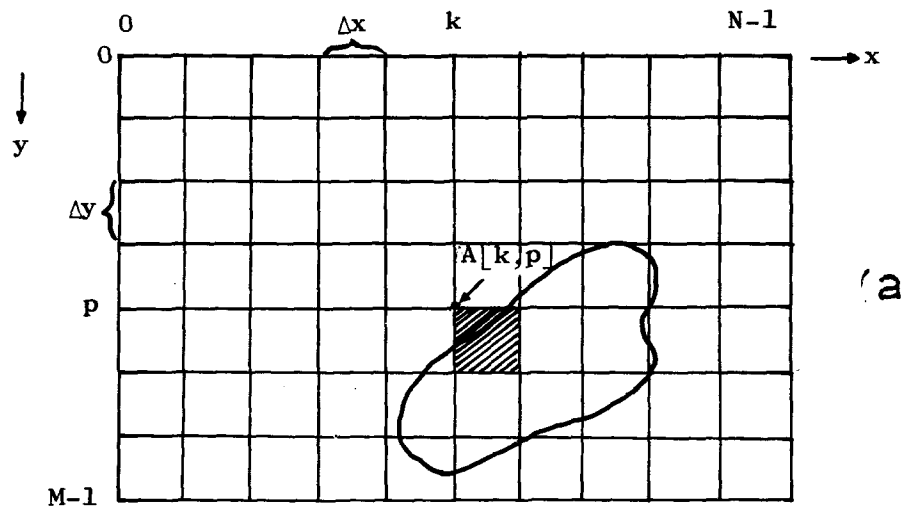


FIG. 1 REPRESENTATION OF DATA MATRIX

- a) The complete matrix
- b) One cell of the matrix

2. PROGRAM DESCRIPTION

A computer program for the drawing of contours, based on the principles outlined in Chapter 1, has been written in the form of an ALGOL procedure, as listed in Appendix A. The purpose of the present chapter is to describe its working method in a little more detail.

Referring to Fig. 1a, the xy-plane can be divided into $(N-1) \times (M-1)$ number of cells. The sides of each cell are labelled as shown in Fig. 1b. A given contour, specified by a constant which is called LEVEL, is then traced in two distinct steps:

STEP 1

The sides of all the $(N-1) \times (M-1)$ cells are inspected to find out which cell is crossed by the contour. This inspection is done by a series of tests of the type

$$(A[k, p] - \text{LEVEL}) \cdot (A[k+1, p] - \text{LEVEL}) \leq 0 . \quad [\text{Eq. 1}]$$

If a particular cell side is crossed, the above test will be true. The results of the tests are stored as a 4-bit integer number in an auxiliary array CELL $[k, p]$. If the North side of the cell is crossed, bit 0 of CELL $[k, p]$ is set to 1; if the West, South or East side is crossed, bit 1, 2 or 3 respectively is set to 1. If the cell is not crossed, the value of CELL $[k, p]$ is set to zero. As an example, the situation indicated in Fig. 1b will give $\text{CELL } [k, p] = 2^0 + 2^1 = 3$ since in this case the North and the West sides are crossed.

Step 1 is terminated when all the elements of the array CELL $[k, p]$ have been set.

STEP 2

After step 1, all information required to trace the contour line from cell to cell is in array CELL [k, p].

In step 2 the contents of this array are inspected sequentially. When an element of the array CELL [k, p] is found to be different from zero the subscripts of this element are stored and the tracing of a new branch of the contour begins, using the information contained in CELL [k, p]. As one proceeds from one cell to the next the value of the corresponding element of array CELL [k, p] is reduced accordingly, by subtracting 1 if the North edge of the cell is crossed, by subtracting 2 if the West edge is crossed, and so on. In normal cases a cell will only have two intersections and after the contour has been traced through a cell, the value of the corresponding matrix element CELL [k, p] is reset to zero, preventing the program from ever returning to that particular cell.

The tracing of a branch of the contour comes to an end either when the contour returns to the cell where the tracing of the branch started, or when the contour crosses the margin of the plot. After that the program returns to the sequential inspection of CELL [k, p] in order to find a new branch of the contour. Step 2 is terminated when all the elements of CELL [k, p] have been inspected.

The contour is drawn as a straight line joining the intersections of the cell sides. The exact position of the intersection is determined by linear interpolation between the values of the adjacent data points.

In the procedure given in Appendix A it is assumed that the sampling is equidistant. However, it is easy to modify the program for using it also with irregular sampling. In this respect it is pointed out that step 1 is completely independent of the sampling and that this applies also to the logic of step 2. Therefore the only modifications required will be to provide the necessary information about the sampling scheme so that the coordinates of the pen movements can be calculated.

In normal cases a cell will be intersected only at two points: on entry and on exit. In a saddle point, however, all four sides of a cell will be crossed and the contour can be drawn in three different ways, all of which could theoretically be valid [Fig. 2].

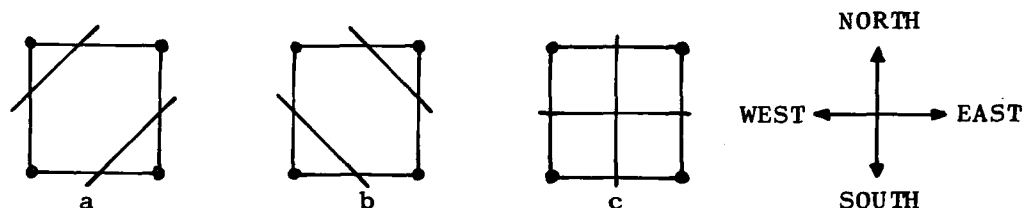


FIG. 2 POSSIBLE WAYS OF CONTOURING AT A SADDLE POINT

Experience with the program, however, has shown that it is not particularly appropriate to allow the contours to cross as in Fig. 2c. This possibility has therefore been removed from the program and at a saddle point the program will always draw the contour lines in the manner shown in Fig. 2a.

When contours are chosen to have exactly the same values as elements of the data matrix it may be found that some cells are intersected at three points. The contouring will then depend on the way the cell is entered and in some conditions can result in a broken contour. The obvious inconveniences of this makes it advisable to choose contour values that differ — if only very slightly — from the matrix values.

3. METHOD OF USE

The ALGOL procedure that has been developed is called by

CONTOUR (A, N, M, XBIAS, YBIAS, LEVEL),

in which the parameters are as follows:

A	Two-dimensional real array A[0:N-1, 0:M-1] containing the data matrix.
N, M	Integers specifying the size of array A.
XBIAS, YBIAS	Real variables for the positioning of the contour plot on the paper.
LEVEL	Real variable specifying the value for the contour to be drawn.

In calling the procedure, a set of contour lines given by the value of LEVEL will be drawn. The position of the plot is such that the subscript 0,0 corresponds to XBIAS, YBIAS. The first subscript of array A refers to the x-direction and the second to the y-direction. The scale for the plotting is presumed to have been set before the procedure is called.

Basically this method of contouring requires that all data points be contained in the N x M matrix A. The procedure will then allocate itself a working array (CELL [k, p]) of the size (N-1) x (M-1). This clearly makes the program less convenient, since all data points have to be available at the same time and since a considerable

amount of memory space is required. When the sample values are not using the full computer word length, the space requirement can be reduced by storing the information otherwise contained in CELL [k, p] in the data matrix itself. This is not implemented on the present procedure, but can easily be done for a particular application.

Another way to overcome the restrictions is to allow for the sectional contouring of a large matrix. In the present procedure this can be accomplished by use of the parameters XBIAS and YBIAS, considering the large matrix as divided — either by lines or columns — into a number of submatrices defined in such a way that there is one line or one column of overlap. The contouring of each of the submatrices can then be done independently, using the parameters XBIAS and YBIAS to position the various sections on the paper.

4. EXAMPLE OF APPLICATION

The described contour program is now used for the presentation of signal processing results, in particular for the display of time-frequency analysis. The output of this analysis is in the form of a two-dimensional matrix, each element of which represents the energy of the signal in a given time and frequency window.

Figure 3 shows an example of this processing. The time signal shown at the bottom of the figure has been filtered by a set of $1/3$ octave filters of the type described in Ref. 2. The output of each of the filters has been squared and integrated over a certain time interval, in this case 15 ms. The upper part of Fig. 3 shows the result of the contouring of this matrix at intervals of 3 dB (arbitrary reference). Thick and thin lines have been used alternately to facilitate the reading of the contour map. If the value of one element of the matrix is known, the contour map shown will contain all the information needed to distinguish maxima from minima.

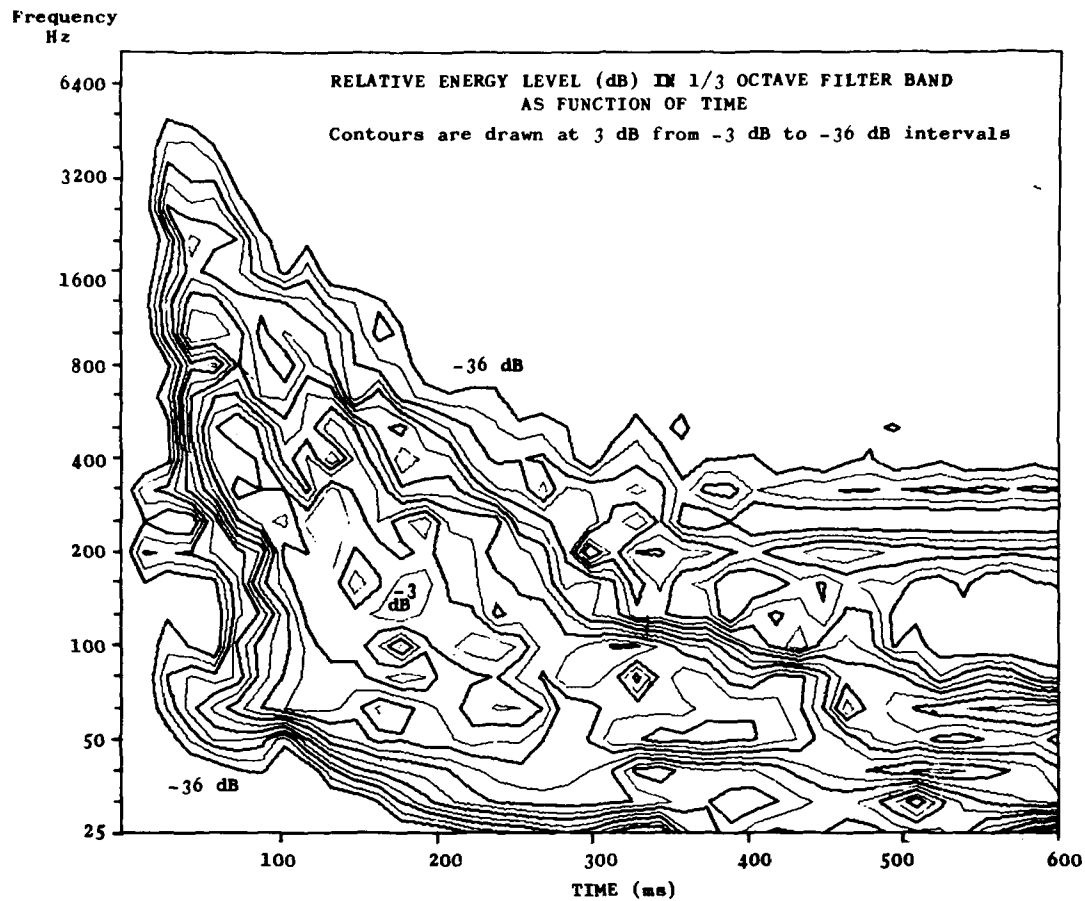


FIG. 3 EXAMPLE OF A CONTOUR MAP

CONCLUSIONS

A method and a computer program for the drawing of contour maps has been described. In this method it is assumed that the data points are available in the form of a two-dimensional matrix, but the whole matrix need not be available at the same time.

Basically the program requires that the sampling be equidistant, but the program can be modified so as to allow also its use when sampling is irregular.

REFERENCES

1. G. Cottafava and G. Le Molt, "Automatic Contour Map".
Communications of the ACM, Vol. 12, No. 7, July 1969,
pp 386-391.
2. J.M. Hovem and M. Thompson, "A Description of Some
Recursive Filters", SACLANTCEN Technical Memorandum No. 167,
June 1971, NATO UNCLASSIFIED.

APPENDIX A

LISTING OF AN ALGOL PROCEDURE FOR CONTOUR MAPPING

The following ALGOL procedure has been developed and used on an ELLIOTT 503 computer, with a CALCOMP plotter as the graphic device. Moving the pen of the plotter to coordinate X,Y is effected by the procedure calls "movepen (X,Y)" and "drawline (X,Y)" for respectively "pen up" and "pen down". The contouring procedure is self-contained in that these are the only two procedures called.

```

procedure CONTOUR(A,N,M,XBIAS,YBIAS,LEVEL);
  value N,M,XBIAS,YBIAS,LEVEL;
  integer N,M;
  real XBIAS,YBIAS,LEVEL;
  array A;
  begin Boolean FIRST,NORTH,WEST,SOUTH,EAST;
    real W1,W2,W3,W4,X,Y;
    integer EXIT,NMAX,MMAX,I1,I2,I3,I4,K,KK,P,PP;
    switch S:=L1,L2,L3,L4,L5,L6;
    integer array CELL[0: N-2,0: M-2];

    procedure STATE(I);
      value I;
      integer I;
      begin I1:=I;
        I4:=I1 div 8;
        I1:=I1-I4*8;
        I3:=I1 div 4;
        I1:=I1-I3*4;
        I2:=I1 div 2;
        I1:=I1-I2*2;
        NORTH:=WEST:=SOUTH:=EAST:=false;
        if I1=1 then NORTH:=true;
        if I2=1 then WEST:=true;
        if I3=1 then SOUTH:=true;
        if I4=1 then EAST:=true;
      end STATE;

    NMAX:=N-2;
    MMAX:=M-2;
    for K:=0 step 1 until NMAX do for P:=0 step 1 until MMAX do
      begin W1:=A[K,P]-LEVEL;
        W2:=A[K+1,P]-LEVEL;
        W3:=A[K,P+1]-LEVEL;
        W4:=A[K+1,P+1]-LEVEL;
        I1:=0;
        if W1*W2 < 0 then I1:=1;
        if W1*W3 < 0 then I1:=I1+2;
        if W3*W4 < 0 then I1:=I1+4;
        if W4*W2 < 0 then I1:=I1+8;
        CELL[K,P]:=I1;
      end;

    for KK:=0 step 1 until NMAX do for PP:=0 step 1 until MMAX do
      begin if CELL[KK,PP]≠0 then
        begin K:=KK;
          P:=PP;
          FIRST:=true;
          STATE(CELL[K,P]);
          if K=0 then goto L2;
          if K=NMAX then goto L4;
          if P=MMAX then goto L3;
          L1: if not (NORTH or WEST or SOUTH or EAST) then goto L6;
          if NORTH then
            begin NORTH:=false;
              EXIT:=1;
              W1:=A[K,P];
              W2:=A[K+1,P];
              goto L5;
            end;
          end;

```

```

L2: if WEST then
begin WEST:=false;
    EXIT:=2;
    W1:=A[K,P];
    W2:=A[K,P+1];
    goto L5;
end;
L3: if SOUTH then
begin SOUTH:=false;
    EXIT:=3;
    W1:=A[K,P+1];
    W2:=A[K+1,P+1];
    goto L5;
end;
L4: if EAST then
begin EAST:=false;
    EXIT:=4;
    W1:=A[K+1,P];
    W2:=A[K+1,P+1];
    goto L5;
end;
goto L1;

L5: CELL[K,P]:=CELL[K,P]-2*(EXIT-1);
W3:=LEVEL-W1;
W4:=W2-W1;
if abs(W4) > 0.001 then W1:=W3/W4 else W1:=0.5;
if EXIT=3 or EXIT=4 then W2:=1 else W2:=0;
if EXIT=1 or EXIT=3 then
    begin X:=K+W1;
        Y:=P+W2;
    end
    else
    begin X:=K+W2;
        Y:=P+W1;
    end;
    X:=X+XBIAS;
    Y:=Y+YBIAS;
    if FIRST then
    begin movepen(X,Y);
        FIRST:=false;
        goto L1;
    end;
    drawline(X,Y);
    if EXIT=1 then P:=P-1 else if EXIT=2 then K:=K-1 else if EXIT=3
    then P:=P+1 else K:=K+1;
    if (K=KK and P=PP) or K < 0 or K > NMAX or P < 0 or P > MMAX
    then goto L6;
    if CELL[K,P]≠0 then
    begin if EXIT=1 then I1:=4;
        if EXIT=2 then I1:=8;
        if EXIT=3 then I1:=1;
        if EXIT=4 then I1:=2;
        CELL[K,P]:=CELL[K,P]-I1;
        STATE(CELL[K,P]);
        goto S[5-EXIT];
    end;
end;
L6:
end;
end CONTOUR;

```